IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SLEEP ALERT SYSTEM FOR DRIVERS

Vedant Sharma
Dept Of Computer Engineering, SOET
D Y PATIL University, Ambi, Pune

Prof. AnupDate
Department of Computer Engineering, SOET
D Y PATIL University, Ambi, Pune

Abstract— **The objective of this project was to create a real-time eye state monitoring system utilizing computer vision techniques and deep learning. The focus was on detecting whether a person's eyes were open or closed, with the goal of improving safety and awareness in different scenarios like driver drowsiness detection and fatigue monitoring.**

The project followed a systematic approach, beginning with the collection and preparation of an eye image dataset. This dataset consisted of images of open and closed eyes, which were resized and normalized for consistency and to facilitate model training. The dataset was then split into training and testing sets for evaluating the model's performance.

Transfer learning was employed using the InceptionV3 model as the base model. Additional layers were added for classification purposes. The weights of the base model were frozen, and the added layers were trained using the prepared dataset. The model was optimized using the Adam optimizer and categorical cross-entropy loss function.

During the training process, data augmentation techniques were utilized, such as rotation, shear, zoom, and shift operations, to increase the diversity and robustness of the training data. The model was trained for multiple epochs, with a validation subset used to monitor performance and prevent overfitting.

Once the model training was completed, a real-time eye monitoring system was implemented. The system utilized a webcam or video input to capture frames. Face and eye detection algorithms were employed to identify the region of interest (ROI) containing the eyes within each frame. The ROI was pre-processed, resized, and normalized before being inputted into the trained model for prediction. Based on the model's output probabilities, the system determined whether the eyes were open or closed.

To improve usability, a scoring mechanism was implemented to track the duration of closed or open eyes. An alarm sound was triggered if the eyes remained closed for an extended period, serving as an alert for potential drowsiness or fatigue.

While the project successfully developed and implemented a real-time eye state monitoring system, its performance depends on the quality of the training data, the effectiveness of the deep learning model, and the

accuracy of the face and eye detection algorithms. Further enhancements could involve incorporating more diverse

and representative datasets, fine-tuning the model architecture, and optimizing the detection algorithms.

Overall, this project demonstrates the potential of computer vision and deep learning techniques in addressing eye state monitoring challenges, providing a foundation for further research and application in various domains requiring improved safety and attention tracking. Overall, the project highlights the potential of computer vision and deep learning techniques in addressing eye state monitoring challenges and offers a foundation for further research and application in various domains requiring enhanced safety and attention tracking [8].

Keywords— machine learning, ANN, CNN, jupyterlab, Keras, haar cascade, classifier, Adam optimizer, RELU, Softmax, Pre-processing, OpenCV, inceptionV3, VGGNet, RESnet

I. INTRODUCTION

A. *What this system is about?*

Driving while fatigued or drowsy poses a significant risk to road safety and can have devastating consequences, including severe accidents, injuries, and loss of life. To address this pressing issue, the

Sleep Alert System for Drivers has been developed as an innovative solution. This system utilizes advanced technology to detect signs of driver fatigue and promptly issue alerts, empowering drivers to prevent accidents caused by drowsiness.

The project involves integrating various components, including camera-based detection systems, data analysis algorithms, and real-time alert mechanisms. By continuously monitoring the driver's behavior, such as eye movements, the Sleep Alert System accurately identifies early indicators of sleepiness and drowsiness.

Upon detecting potential sleepiness or drowsiness, the system triggers immediate alerts to the driver, effectively notifying them of their deteriorating state. These alerts are delivered through auditory signals, ensuring drivers are promptly informed and motivated to take necessary action.

The Sleep Alert System for Drivers goes beyond detection and alerting; it also prioritizes driver safety. Leveraging data analysis and machine learning techniques, the system can determine if the driver is already asleep or on the verge of falling asleep due to fatigue. By providing auditory signals, it assists drivers in making informed decisions regarding rest breaks and overall well-being [9][10].

In summary, the Sleep Alert System for Drivers represents an innovative approach to mitigating the risks associated with drowsy driving. By employing cutting-edge technology and comprehensive monitoring, it equips drivers with the tools they need to enhance their safety and prevent accidents caused by fatigue or sleepiness.

B. Necessity of this system:-

The urgency of developing and implementing effective measures to combat the dangers of drowsy or sleepy driving cannot be overstated. The Sleep Alert System for Drivers emerges as a crucial solution to detect and prevent accidents caused by drowsiness.

The need for such a system is underscored by the alarming statistics surrounding drowsy driving incidents and their impact on road safety. In 2021 alone, road accidents claimed the lives of 153,972 individuals and caused harm to 384,448 people, as reported by the Ministry of Road Transport and Highways Transport Research Wing. Shockingly, the age group most affected by these accidents is between 18 and 45 years old, accounting for nearly 67 percent of all accidental deaths. [1] These distressing figures emphasize the dire need to tackle this issue and implement preventive measures effectively.

Moreover, the consequences of accidents resulting from drowsy driving are often severe. Falling asleep at the wheel leads to loss of control, high-speed collisions, rollovers, and collisions with other vehicles or objects. These accidents result in life-altering injuries and tragic loss of life. By adopting the Sleep Alert System, the objective is to proactively prevent such accidents, thereby saving lives and minimizing the devastating impact on individuals, families, and communities.

Another critical factor that highlights the necessity of the Sleep Alert System is the inherent limitations of human judgment and self-awareness when it comes to recognizing fatigue. Sleepiness or drowsiness can set in unexpectedly, and drivers often underestimate their level of fatigue. Many individuals mistakenly believe they can push through their tiredness or rely on brief moments of microsleep to continue driving. Unfortunately, these risky behaviors significantly increase the likelihood of accidents. The Sleep Alert System acts as an objective and reliable monitor, identifying signs of sleepiness or drowsiness that may go unnoticed by the driver. It provides timely alerts, ensuring drivers are aware of their fatigue levels and enabling them to take appropriate action to prevent accidents.

Furthermore, it is essential to note that the Sleep Alert System is not limited to specific driver categories, such as long-haul truck drivers or commercial vehicle operators. Fatigue can affect any driver, including those commuting to work, individuals on long road trips, or drivers working irregular shifts. The system has the potential to benefit all drivers, regardless of their driving patterns or vehicle types. By providing a universal solution to address drowsy driving, the Sleep Alert System contributes to the improvement of road safety for everyone.

In conclusion, the development and implementation of the Sleep Alert System for Drivers is an imperative response to the dangers of drowsy driving. By proactively detecting signs of sleepiness or drowsiness and issuing timely alerts, this system aims to prevent accidents, save lives, and enhance road safety for all drivers.

C. Objective of the system:-

The Sleep Alert System for Drivers aims to enhance road safety by effectively detecting and preventing drowsy driving incidents. The system is designed with the following key objectives in mind:

1) ***Early Detection of Sleepiness:*** The primary goal of the Sleep Alert System is to detect signs of sleepiness or drowsiness in drivers at an early stage. By monitoring indicators like eye movements, the system can accurately identify the onset of fatigue. This early detection allows for timely intervention, triggering alarms to prevent the progression of drowsiness and reducing the risk of accidents.

2) ***Timely Alerts and Warnings*:** Once signs of drowsiness are detected, the system's secondary objective is to promptly alert the driver. Through auditory signals, the system notifies the driver of their deteriorating state, urging them to take immediate action. By providing timely warnings, the system aims to prevent accidents by ensuring drivers remain vigilant and responsive while on the road.

*3) **Universal Application:*** The Sleep Alert System is designed to have a universal application across various driving contexts. It is not limited to specific driver categories but intended to benefit all drivers, whether they are commercial drivers or individuals driving for personal or recreational purposes. Whether it's a long-haul trucker, a daily commuter, or a family on a road trip, the system strives to enhance safety for drivers of all types, irrespective of their driving patterns or vehicle types.

*D. **Theme of the system:-***

Theme: "Vigilance for Safer Journeys"

The theme for the Sleep Alert System for Drivers project is "Vigilance for Safer Journeys." This theme encapsulates the core objective of the system, which is to enhance road safety by promoting driver vigilance and preventing incidents of drowsy driving.

The chosen theme emphasizes the crucial importance of remaining vigilant while embarking on journeys, as drowsiness can severely impair a driver's ability to react and make critical decisions. By highlighting the value of vigilance, the project aims to raise awareness among drivers regarding the hazards associated with driving while fatigued, thereby encouraging proactive measures to ensure safer journeys.

Moreover, the theme reflects the pivotal role played by the Sleep Alert System in fostering vigilance. Through the utilization of advanced technology, the system enables drivers to maintain an alert and attentive state by detecting early signs of drowsiness and issuing timely alerts. By underscoring the significance of vigilance, the project encourages drivers to prioritize their well-being and take necessary actions to prevent accidents resulting from fatigue.

Furthermore, the theme conveys the project's commitment to cultivating a safety-oriented culture on the roads. By promoting vigilance and advocating for safer journeys, the Sleep Alert System aims to contribute to a positive shift in driver behavior and attitudes towards drowsy driving. It encourages drivers to adopt a proactive approach to managing their fatigue, make informed decisions about rest breaks, and implement strategies to prevent accidents.

*E. **Intended Audience:*** The intended audience for this document comprises the development team, the project evaluation jury, and other technology enthusiasts interested in furthering the project's work.

*F. **Problem Definition:***

The Sleep Alert System for Drivers addresses the problem of drowsy driving, which poses a significant risk to road safety. The objective is to detect signs of driver fatigue early and issue timely alerts to prevent accidents caused by drowsiness.

Drowsy driving is a prevalent issue as drivers often underestimate its dangers and fail to recognize their own fatigue until it is too late. This lack of awareness increases the risk of accidents and jeopardizes lives.

The problem stems from drivers' difficulties in accurately assessing their level of fatigue and disregarding warning signs. This leads to lapses in attention, slower reaction times, and even episodes of microsleep, impairing driving abilities and heightening the likelihood of accidents.

To address this problem, the Sleep Alert System integrates sensors, data analysis, and alert mechanisms to objectively detect fatigue. Challenges include developing accurate algorithms, seamless sensor integration, and implementing attention-grabbing alerts.

By tackling these challenges, the system aims to mitigate the risks associated with drowsy driving, prevent accidents, and save lives. It empowers drivers to prioritize their well-being, remain vigilant, and make informed decisions regarding driving and rest breaks.

II. REVIEW OF LITERATURE

*A. **Technologies Used:-***

Here is a small description of all the libraries we used from the starting to the end of the project

*1) **Jupyter lab:***

Jupyter Lab is an interactive development environment (IDE) that enables users to create and share documents containing live code, visualizations, and explanatory text. It provides a flexible and user-friendly interface for working with Jupyter notebooks, which combine executable code with narrative elements. Jupyter Lab supports various features such as a file explorer, code editor, and interactive consoles, making it a preferred choice for data analysis, machine learning, and scientific computing workflows.

*2) **Python:*** Python is a versatile programming language known for its simplicity and readability. It offers a clean and concise syntax, making it easy to learn and understand. Python supports multiple programming paradigms and has an extensive standard library. Its ecosystem includes a wide range of tools and frameworks. Python is widely used in web development, data analysis, scientific computing, and artificial intelligence applications. Its cross-platform compatibility and integration capabilities make it a popular choice among developers [15].

3) Python Modules/Libraries:

a) Data Pre-Processing:

- OS module: The OS module provides functions for interacting with the operating system, offering utilities for file operations and directory manipulation.
- Shutil module: The Shutil module allows high-level file operations such as copying, creating, and deleting files and directories.
- Glob module: The Glob module is used for file pattern matching and searching files with specific patterns or names.
- Random module: The Random module provides functions for generating random numbers and values in Python.

b) **Model Training**:

- TensorFlow: TensorFlow is a Python library for fast numerical computing, particularly suitable for machine learning and deep learning tasks[3].
- Keras: Keras is a high-level deep learning API that simplifies the process of building and training neural networks. It integrates seamlessly with TensorFlow[12].
- Inceptionv3: Inception v3 is a well-known image recognition model that achieves high accuracy on the ImageNet dataset.

c) Model Implementation:

- Cv2 (OpenCV): OpenCV is a computer vision library that offers a wide range of functionalities for image and video processing.
- Load Model: A library used for loading pre-trained models.
- NumPy: NumPy is a fundamental library for numerical computations in Python, providing support for multi-dimensional arrays and mathematical operations.
- Mixer: A library used for generating instances of Django or SQL Alchemy models, commonly utilized for testing and fixture replacement [18].

4) **Operating System:**

- Windows 11 (used)

B. **Hardware Requirements:**

- Processor: Core i5 4590
- RAM: 8 GB
- GPU: GTX1060 6 GB [7]
- Webcam

*C. **Existing system: -***

*1) **DMS by TESLA***

Tesla, a prominent electric vehicle manufacturer, has integrated a drowsiness alert system into its vehicles to enhance driver safety and mitigate the risks associated with drowsy driving. Known as the Driver Monitoring System (DMS), this advanced system employs cutting-edge sensor technology and artificial intelligence algorithms.

The DMS in Tesla vehicles utilizes a combination of strategically placed cameras and sensors within the cabin. These sensors continually monitor various aspects of the driver's behavior, including eye movements, head position, and facial expressions. By analyzing these parameters, the system detects potential signs of driver fatigue or inattention.

When signs of drowsiness or inattention are detected, the DMS promptly issues visual and audible alerts to draw the driver's attention back to the road. These alerts can be presented through visual cues on the instrument cluster or center display, accompanied by audio warnings.

In addition to real-time alerts, Tesla's DMS provides recommendations for the driver to take a break and rest if signs of fatigue persist. These recommendations prioritize the driver's well-being and aim to minimize the inherent risks associated with drowsy driving.

Notably, Tesla's DMS continually evolves through over-the-air software updates, enabling ongoing refinements and enhancements to its algorithms and detection capabilities. This adaptive approach allows the system to become increasingly accurate and reliable over time, adapting to diverse driving conditions and individual driver behaviors.

The integration of the drowsiness alert system within Tesla vehicles aligns with the company's unwavering commitment to driver safety and their vision of advancing autonomous driving technologies. By actively monitoring driver attention and intervening when drowsiness or distraction is detected, Tesla strives to reduce the likelihood of accidents caused by driver fatigue, ultimately contributing to safer roads for all users.

*2) **Drawback of the Existing System: -***

While the incorporation of the Drowsiness Alert System in Tesla vehicles yields substantial advantages in terms of driver safety, it is crucial to acknowledge certain potential drawbacks associated with its implementation. These drawbacks are as follows:

a) Limitations in Challenging Conditions: The efficacy of the DMS may be compromised in adverse driving conditions, including heavy rain, fog, or intense glare. Factors such as reduced visibility or environmental interferences can influence the system's ability to accurately detect drowsiness or driver inattention, which may lead to an increased risk of false negatives.

b) Cost: The manufacturing costs and product availability pose significant considerations that impact the widespread adoption of the system. The high production costs associated with implementing the Drowsiness Alert

System may impede its accessibility and affordability for consumers.

These potential drawbacks should be acknowledged and carefully evaluated when assessing the overall effectiveness and feasibility of the Drowsiness Alert System in Tesla vehicles. Further research and development efforts should focus on addressing these limitations to ensure optimal performance and maximize the benefits of the system for driver safety.

III. METHODOLOGY

A. *System Architecture and Components:-*

Worldwide, road accidents are often caused by tired and sleepy drivers. Recent figures from organizations that promote road safety show that sleepy driving is a key contributing factor in many incidents. Drowsy driving is thought to be responsible for approximately 70,000 accidents and 800 fatalities annually in the United States alone. These startling figures demonstrate the urgent need for efficient initiatives to reduce the hazards posed by driver tiredness.

1) *Numerical Data on Accidents:-*

To emphasize the severity of the problem, empirical data can be included to provide a quantitative understanding of accidents caused by sleep-related factors. For example, citing statistics from national or international road safety organizations, you can present the total number of accidents, injuries, and fatalities attributed to drowsy driving in a given period. These figures may vary depending on the region or country under consideration.

For instance, in a study conducted by the National Highway Traffic Safety Administration (NHTSA) in the United States, it was reported that drowsy driving accounted for approximately 2.5% of fatal crashes and 2% of injury crashes [2]. Furthermore, the study estimated that around 6,000 fatal crashes each year are due to drowsy driving, resulting in an estimated 1,550 deaths and 71,000 injuries.

Including such numerical data further emphasizes the gravity of the problem and provides a factual basis for the urgent need to address drowsy driving through effective countermeasures

2) *System Architecture:-*

The Sleep Alert System for Drivers adopts a comprehensive architecture that integrates hardware and software components to monitor and alert drivers of potential sleepiness. The architecture consists of the following key components:

a) *Data Acquisition:* The system utilizes various sensors and data collection devices to gather relevant physiological signals from the driver. These sensors may include heart rate monitors, accelerometers, eye trackers, or brain activity sensors.

b) *Signal Processing:* The collected data is processed using signal processing techniques to extract meaningful features that indicate driver fatigue or drowsiness. This may involve algorithms for analyzing heart rate variability, eye movement patterns, or changes in steering behavior.

c) *Sleep Detection Algorithm:* The system employs sophisticated machine learning algorithms or pattern recognition techniques to classify the driver's alertness level. These algorithms utilize the extracted features to detect patterns indicative of drowsiness or fatigue.

d) *Alert Generation:* When the sleep detection algorithm identifies signs of drowsiness, the system generates alerts to notify the driver and prevent potential accidents. The alerts may include audible warnings, visual cues, or haptic feedback, depending on the design and implementation of the system.

e) *User Interface:* The system incorporates a user interface that provides real-time feedback to the driver. This interface may include a dashboard display, mobile application, or heads-up display (HUD) to convey relevant information about the driver's alertness level and the system's recommendations.

B. *Data Preprocessing and Feature Extraction:-*

1) *Data Preprocessing:-*

In the data preprocessing step, we performed necessary tasks on the "Mrleyes" dataset, which consists of thousands of images. Our aim was to prepare the data for effective use in our system.

Initially, we extracted the names of each image in the dataset using a for loop. This helped us create a list of image names, facilitating easy access to the images. Next, we divided the data into two sets: train and test. We did this separately for images with open eyes and images with closed eyes. This division allowed us to assess the system's performance accurately and ensure its ability to handle new, unseen data.

These data preprocessing steps laid the groundwork for the subsequent feature extraction process. Feature extraction involves extracting relevant information or characteristics from the images. By organizing the data and dividing it into train and test sets, we created a structured framework to extract meaningful features from the images, which are vital for our system's functioning.

Data preprocessing is a crucial step in machine learning and computer vision projects. It ensures that the data is properly organized, labeled, and divided for optimal model training and evaluation. By simplifying and preparing the data, we set the stage for effective analysis and utilization in our system.

2) *Feature Selection:-*

In the feature selection step, we employed various techniques to extract relevant features from the images in the "Mrleyes" dataset. These features play a crucial role in characterizing and distinguishing between open and closed eyes.

To begin, we utilized an image data generator to perform resizing operations. By rescaling the images to a standardized size of 80x80 pixels, we ensured uniformity and consistency in the dataset. Rescaling helps in reducing computational complexity and normalizing the images for further analysis.

Additionally, we incorporated Haar cascades and cascade classifiers as part of the feature selection process. Haar cascades are machine learning-based algorithms that detect specific patterns in images. We employed pre-trained Haar cascades specifically designed for eye detection. These cascades enable us to identify the regions of the image containing eyes accurately.

By using cascade classifiers, which are trained on positive and negative samples, we can distinguish between open and closed eyes based on predefined features and thresholds. The cascade classifier helps in categorizing the eye regions detected by the Haar cascades into open or closed classes.

The combined use of image rescaling, Haar cascades, and cascade classifiers facilitates effective feature selection. These techniques enable us to extract informative features from the images, making them suitable for subsequent analysis and classification tasks within our system.

Feature selection is a critical step in machine learning and computer vision, as it helps identify the most relevant and discriminative features for accurate classification or analysis. The incorporation of image data generators, Haar cascades, and cascade classifiers enhances the quality and suitability of the selected features for the task at hand [17].

C. *Model Selection and Training:-*

1) *Model selection:-*

In the model selection process, we evaluated different models to determine the most suitable one for our task of classifying open and closed eyes in the "Mrleyes" dataset. Among the models considered, we chose the Inception V3 model as our primary choice, but we also explored alternative models. Here's an explanation of why we selected the Inception V3 model and considered other options

a) *Inception V3 Model:* We opted for the Inception V3 model due to its excellent performance and versatility in image classification tasks. Inception V3 is a deep convolutional neural network (CNN) architecture that has been widely used and proven effective for various computer vision tasks. It leverages a combination of convolutional layers, pooling, and inception modules to extract complex and hierarchical features from images. The model has a large number of parameters, enabling it to learn intricate patterns and representations [6].

2) *Why Inception V3:-*

a) *High Accuracy*: Inception V3 has demonstrated impressive accuracy in image classification benchmarks, making it suitable for our task of distinguishing between open and closed eyes.

b) *Pre-trained Model*: Inception V3 is readily available as a pre-trained model on large-scale image datasets like ImageNet. This pre-training helps to capture general image features and accelerate the learning process on our specific eye classification task.

c) T*ransfer Learning:* Leveraging transfer learning, we can fine-tune the Inception V3 model on our dataset, benefiting from the knowledge learned from a vast amount of diverse images

3) *Consideration of Alternative Models:-*

While we selected Inception V3 as our primary choice, we also considered other models to explore different architectures and evaluate their performance. Alternative models could include popular CNN architectures like ResNet, VGGNet, or MobileNet, which have been widely adopted in computer vision tasks. Assessing multiple models allows us to compare their performance, considering factors such as accuracy, computational efficiency, and memory requirements.

The model selection process involves weighing various factors such as accuracy, complexity, computational resources, and suitability for the specific task. The choice of Inception V3 as our primary model was driven by its established performance in image classification, pre-trained availability, and potential for transfer learning. By considering alternative models, we ensure a comprehensive evaluation of the available options before finalizing our selection.

4) *Training :-*

In the training step of our eye classification task, we employed the Inception V3 model along with various techniques and tools to train and optimize the model's performance. Here's an explanation of the training process and the techniques used:

a) *Inception V3 Model:* We utilized the Inception V3 model as the backbone architecture for our eye classification task. Inception V3 is a deep CNN architecture known for its ability to capture intricate image features and patterns. We leveraged its pre-trained weights as a starting point, which were learned from a large-scale dataset[5].

b) Enhance model generalization, we employed the ImageDataGenerator from TensorFlow. This data augmentation technique generated additional training samples by applying random transformations such as rotations, zooms, and flips to the images. Data augmentation helps prevent overfitting and improves the model's ability to handle variations in real-world data [4].

c) *Activation Functions*: We used Rectified Linear Unit (ReLU) activation functions within the Inception V3 model. ReLU helps introduce non-linearity, allowing the model to learn complex mappings between the input data and the desired output. Additionally, the Softmax activation function was applied to the final layer of the model to produce class probabilities, enabling us to classify the input images into open or closed eyes.

d) *Early Stopping:* To prevent overfitting and determine the optimal number of training epochs, we employed early stopping. This technique monitors a specified metric, such as validation loss, and stops training when the metric fails to improve for a certain number of epochs. Early stopping helps prevent the model from memorizing the training data and ensures better generalization on unseen data.

e) *Adam Optimizer:* We used the Adam optimizer, a popular optimization algorithm, to update the model's weights during training. Adam combines the benefits of adaptive learning rate methods and momentum, enabling efficient convergence during gradient descent and speeding up the training process [11].

f) *ReduceLROnPlateau:* We implemented the ReduceLROnPlateau callback in TensorFlow. This technique dynamically reduces the learning rate when the validation loss plateaus, allowing the model to fine-tune its parameters more accurately during later training stages.

Through the training step, we iteratively fed the augmented and preprocessed training data to the Inception V3 model. The model learned to classify the eye images into open or closed categories by minimizing the loss function using the Adam optimizer. Early stopping and learning rate reduction techniques were employed to prevent overfitting and enhance training efficiency [11].

By leveraging these techniques and tools, we aimed to train the Inception V3 model effectively, optimize its performance, and improve its ability to accurately classify open and closed eyes in real-world scenarios.

D. *Performance Evaluation:*

TABLE I. THIS IS THE TRUTH TABLE FOR THE MODEL

Eyes state	**Prediction [0][0]**	**Prediction [0][1]**	**Score**	**Action**
Closed	>0.20	-	Increase score by1	
				Play emergency alarm sound
Open	-	>0.90	Decrease score by 1	
			<0	Reset score to 0

1) *Explanation:*

The code captures video frames from a source (specified by `cap`) and processes them.
It detects faces and eyes within each frame using Haar cascades (`face_cascade` and `eye_cascade`).
For each detected eye region, it performs the following steps:

- Preprocesses the eye region by resizing it to (80, 80) and normalizing pixel values.
- Feeds the preprocessed eye region to the `model` for prediction.
- Checks the prediction values (`prediction[0][0]` and `prediction[0][1]`) to determine if the eyes are closed or open.
- Based on the prediction, it updates the score and displays the corresponding text on the frame using OpenCV functions.

- If the score exceeds a threshold (`Score > 15`) and the emergency alarm sound file is provided (`sound`), it plays the sound.

- The frame with the annotations is displayed using OpenCV's `imshow` function.

- The loop continues until the user presses the 'q' key to exit.

- Finally, the video capture is released (`cap.release()`) and OpenCV windows are closed (`cv2.destroyAllWindows()`).

Note: - That the behavior of this code heavily depends on the specific model used for eye state classification, the quality of the trained model, and the accuracy of the face and eye detection using Haar cascades [14].

2) ***Summary***:-

The table summarizes the behavior and actions taken based on the predicted state of the eyes and the corresponding prediction scores. Here's a summary of the table:

a) Eye State:

- Closed: The predicted state of the eyes is closed.
- Open: The predicted state of the eyes is open.

b) ***Prediction***:

- [0][0]: If the eyes are closed, the prediction score is greater than 0.20.
- [0][1]: If the eyes are open, the prediction score is greater than 0.90.

c) ***Actions***:

- Closed Eyes and Score > 0.20: If the eyes are closed and the prediction score exceeds 0.20, the score is increased by 1. Additionally, an emergency alarm sound is played to alert the driver.
- Open Eyes and Score > 0.90: If the eyes are open and the prediction score exceeds 0.90, the score is decreased by 1.
- Open Eyes and Score < 0: If the eyes are open and the prediction score falls below 0, the score is reset to 0.

The summary provides an overview of the actions triggered based on the predicted state of the eyes and the associated prediction scores. These actions aim to respond appropriately to the detected eye state and ensure driver safety.

IV. SYSTEM IMPLEMENTATION

A. Hardware and Software Setup:

1) Hardware Setup:

The hardware setup for the Sleep Alert System consists of the following components:

a) *Webcam*: A webcam is utilized to capture real-time video feed of the driver's face and eyes. It serves as the input device for the system to monitor the driver's alertness and detect signs of drowsiness or eye closure.

b) *Intel Core i5 10th Gen Processor:* The system is powered by an Intel Core i5 10th generation processor. This processor provides sufficient computing power to handle the image processing and analysis tasks involved in real-time monitoring of the driver's eyes.

c) *Nvidia GeForce GTX 1650:* The Sleep Alert System utilizes an Nvidia GeForce GTX 1650 graphics card. This dedicated GPU offers hardware acceleration and parallel processing capabilities, enabling faster and more efficient image and video processing tasks. It assists in real-time analysis of the webcam feed and facilitates smoother execution of the system's algorithms.

d) *8GB RAM:* The system is equipped with 8GB of Random Access Memory (RAM). Sufficient RAM capacity ensures smooth multitasking and efficient handling of the data processed during the monitoring and analysis tasks. It helps in maintaining system responsiveness and supports the timely execution of the sleep detection algorithms.

e) *1TB Memory*: The system includes a 1TB hard drive, providing ample storage space for storing the necessary software components, datasets, and additional resources required for the Sleep Alert System.

The combination of a webcam, Intel Core i5 10th Gen processor, Nvidia GeForce GTX 1650 graphics card, 8GB RAM, and 1TB memory forms a robust hardware setup for the Sleep Alert System. This configuration enables real-time video processing, efficient algorithm execution, and ensures smooth performance for accurate drowsiness detection and alerting mechanisms.

2) Software Setup:

The software setup for the Sleep Alert System involves the integration of various software components to enable real-time monitoring of driver alertness and drowsiness detection. Here's an overview of the software setup:

a) *Operating System*: The system is built on an operating system such as Windows, macOS, or Linux, providing a stable and reliable foundation for running the software components.

b) *Python:* The Sleep Alert System is developed using the Python programming language. Python offers a wide range of libraries and frameworks for image processing, machine learning, and computer vision, making it suitable for implementing the system's algorithms and functionalities.

c) *OpenCV:* OpenCV (Open-Source Computer Vision Library) is a widely used open-source library for computer vision tasks. It provides extensive functionality for image and video processing, making it suitable for face detection, eye tracking, and feature extraction in the Sleep Alert System [18].

d) *TensorFlow/Keras:* TensorFlow and Keras are popular deep learning frameworks that enable the implementation of complex machine learning models. These frameworks provide tools for training and deploying neural networks, allowing the Sleep Alert System to use pre-trained models or train custom models for eye state classification [12].

e) *Haar Cascade and Cascade Classifier:* Haar Cascade is a machine learning-based object detection method used for identifying specific objects or features in images. Cascade Classifier is an algorithm derived from Haar Cascade, primarily used for real-time object detection. In the Sleep Alert System, Haar Cascade and Cascade Classifier are employed to detect facial features, specifically the eyes, for monitoring eye closure and drowsiness [17].

f) *Data Preprocessing:* The system incorporates data preprocessing techniques to prepare the input data for analysis and model training. This involves resizing and normalizing the captured images or video frames to a standard size, enhancing image quality, and extracting relevant features for further processing.

g) *Model Training and Evaluation:* The Sleep Alert System utilizes machine learning models to classify the eye state (open or closed). The models are trained using labeled datasets, and their performance is evaluated using metrics such as accuracy, precision, and recall. The models can be fine-tuned and optimized to improve their performance in real-world scenarios.

h) *Alarm and Alerting Mechanism:* The software includes an alarm and alerting mechanism that triggers an alert when drowsiness is detected. This can include playing an alarm sound, displaying a warning message, or sending notifications to the driver or relevant authorities.

The software setup combines various tools, libraries, and frameworks to enable real-time monitoring, analysis, and detection of driver drowsiness in the Sleep Alert System. The integration of these software components allows for efficient processing, accurate classification, and timely alerting to ensure driver safety on the road.

B. Data Collection and Processing:

1) ***Data Collection:***

The data collection process for the Sleep Alert System involved gathering a diverse dataset from the MRL Eye Database. The dataset consisted of 15,000 pupil points obtained from various eye samples. The collected data encompassed a wide range of scenarios, including male and female eyes, eyes with and without glare, open and closed eyes, eyes with and without spectacles, and eyes captured under different lighting conditions.

The purpose of collecting such a diverse dataset was to ensure the robustness and effectiveness of the Sleep Alert System across various real-world situations. By incorporating different eye characteristics and environmental factors, the system can better adapt to different individuals, lighting conditions, and eye-related variations.

During the data collection process, careful consideration was given to capturing representative samples to cover a broad spectrum of eye appearances and conditions. This approach helps the system generalize well and make accurate predictions when presented with new eye samples.

By incorporating male and female eyes, the system can account for any potential gender-specific variations in eye characteristics. The inclusion of eyes with and without glare helps the system handle situations where drivers may experience glare from oncoming headlights or other light sources. Eyes captured with spectacles and without spectacles account for individuals who wear glasses or contact lenses, ensuring the system's compatibility with individuals who have corrective eyewear. Additionally, gathering data under various lighting conditions allows the system to adapt to different illumination levels commonly encountered during driving, such as daytime, nighttime, or low-light conditions.

The collected dataset serves as a valuable resource for training and testing the Sleep Alert System's algorithms. By leveraging this diverse dataset, the system can learn and identify patterns specific to different eye characteristics, making it capable of accurately detecting drowsiness and alerting drivers in real-time.

Overall, the data collection process focused on capturing a comprehensive range of eye samples, considering factors such as gender, glare, eye state (open or closed), spectacles usage, and lighting conditions. This diverse dataset plays a crucial role in training and enhancing the accuracy of the Sleep Alert System, ultimately contributing to improved driver safety on the roads.

2) ***Processing:***

The data processing stage in the Sleep Alert System involves several important steps to prepare the collected eye data for analysis and model training. Here is an overview of the data processing steps:

a) *Data Cleaning:* The collected eye data may contain noise, outliers, or inconsistencies that could adversely affect the system's performance. Data cleaning techniques, such as removing duplicate entries, handling missing values, and filtering out irrelevant or corrupted data, are applied to ensure the data is of high quality and integrity.

b) *Image Preprocessing:* The eye images captured in the dataset may require preprocessing to enhance their quality and standardize their appearance. Techniques like resizing the images to a consistent resolution (e.g., 80x80 pixels), adjusting brightness and contrast, and applying image filters may be employed to improve the image quality and facilitate accurate feature extraction.

c) *Feature Extraction:* Extracting meaningful features from the eye images is a crucial step in the Sleep Alert System. Feature extraction techniques, such as Haar cascade and cascade classifiers, can be utilized to identify and extract specific eye-related features, such as eye contours, edges, and texture information. These extracted features serve as inputs to the subsequent stages of the system, enabling accurate classification of eye states [17].

d) *Data Augmentation:* To enhance the robustness and generalization capabilities of the system, data augmentation techniques may be applied. These techniques involve generating additional training samples by applying transformations such as rotation, scaling, flipping, and adding noise to the existing eye images. Data augmentation helps the model handle variations in pose, lighting, and other factors, improving its ability to accurately detect drowsiness in different scenarios.

e) *Train-Test Split:* The processed eye data is typically divided into separate training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance. The data is split carefully to ensure a representative distribution of eye states (open or closed) and other relevant factors across both sets, enabling reliable model assessment.

f) *Normalization:* Normalization is applied to scale the feature values to a common range, typically between 0 and 1. This step helps in optimizing the learning process of the model and ensures that features with different scales do not disproportionately influence the training process.

g) *Model Input Formatting*: The processed eye data is prepared in a suitable format to be fed into the selected machine learning model. This involves structuring the data into appropriate input tensors or arrays that the model expects, considering factors such as batch size and input dimensions.

By performing these data processing steps, the Sleep Alert System ensures that the collected eye data is clean, standardized, and enriched with relevant features. This processed data forms the foundation for training the machine learning model, enabling accurate classification of eye states and effective detection of drowsiness in real-time driving scenarios.

C. ***Real-Time Sleep Detection and Alert Generation:***

Real-time sleep detection in the Sleep Alert System involves a series of steps using OpenCV to capture the source video, apply a cascade classifier for eye state classification, and generate a score. Here's an overview of the process:

1) *Video Capture:* The system utilizes OpenCV's video capture functionality to access the live video feed from a webcam or a recorded video source. OpenCV provides a convenient interface to capture frames from the video source in real-time [16].

2) *Frame Processing:* Each frame from the video feed is processed individually to extract the relevant region of interest (ROI) containing the driver's eyes. OpenCV's image processing capabilities are leveraged to isolate and enhance the eye region, ensuring better accuracy in subsequent steps.

3) *Cascade Classifier:* A pre-trained cascade classifier, such as Haar cascade, is applied to the extracted eye region. The cascade classifier is trained to recognize specific patterns associated with open and closed eyes. By passing the eye region through the cascade classifier, the system can determine the eye state as open or closed [16][17].

4) *Eye State Classification:* The cascade classifier analyzes the eye region and

classifies it as open or closed based on the trained patterns it recognizes. The classification result is used to update the system's internal score.

5) *Score Management:* The system maintains a score that reflects the driver's level of drowsiness. The score is incremented when the classifier detects closed eyes, indicating drowsiness. Conversely, the score is decremented when open eyes are detected, suggesting alertness. This scoring mechanism provides a continuous assessment of the driver's state of alertness.

6) *Threshold Evaluation:* The system continuously evaluates the score against a predefined threshold, typically set to 15 in this case. If the score exceeds the threshold, it indicates a high probability of drowsiness. In such cases, an alarm sound is triggered as a warning signal to alert the driver and prevent potential accidents.

By integrating OpenCV's video capture capabilities, cascade classifiers, and score management, the Sleep Alert System can perform real-time sleep detection. This approach allows for the continuous monitoring of the driver's eye state and timely intervention in case of drowsiness, ensuring enhanced safety on the roads.

D. Evaluation Metrics and Performance Analysis

TABLE II. CONFUSION MATIRX

	Predicted Closed	Predicted Open
Actual Closed	True Positive	False Negative
Actual Open	False Positive	True Negative

The table you provided is called a confusion matrix or an error matrix. It is used to evaluate the performance of a classification model by comparing the predicted labels with the true labels. The table is organized as follows:

- True Positive (TP): The model correctly predicted the positive class (Closed) when the actual label was also positive (Closed).
- False Negative (FN): The model incorrectly predicted the negative class (Open) when the actual label was positive (Closed).
- False Positive (FP): The model incorrectly predicted the positive class (Closed) when the actual label was negative (Open).
- True Negative (TN): The model correctly predicted the negative class (Open) when the actual label was also negative (Open).

The confusion matrix helps to quantify the performance of a classification model by providing insights into the types of errors it makes. From the confusion matrix, various evaluation metrics can be calculated, such as accuracy, precision, recall, and F1 score, which provide a more comprehensive assessment of the model's performance.

E. User Feedback and System Usability Evaluation

User feedback and system usability evaluation are essential components of assessing the effectiveness and user satisfaction of a system. They help in identifying areas for improvement and ensuring the system meets the needs and expectations of its users. Here's an explanation of user feedback and system usability evaluation:

1) User Feedback:

User feedback involves collecting input and opinions from users regarding their experience with the system. This can be obtained through various methods such as surveys, interviews, or user feedback forms. The feedback may cover aspects such as system performance, ease of use, accuracy, and any issues or suggestions users may have. It provides valuable insights into user perspectives and helps identify areas that require attention or enhancement.

User feedback can be collected at different stages of system usage, such as during initial testing, after system updates, or as part of regular user satisfaction assessments. It allows developers and designers to understand user needs, preferences, and pain points, ultimately guiding system improvements and updates to enhance user satisfaction.

2) System Usability Evaluation:

System usability evaluation aims to assess the overall usability and user experience of the system. It involves conducting usability tests and evaluations to measure how well the system meets user requirements and how easily users can accomplish their tasks.

Usability evaluations typically involve the following steps:

a) Defining usability objectives: Establishing specific usability goals and metrics to evaluate the system against.
b) Selecting evaluation methods: Choosing appropriate evaluation techniques, such as heuristic evaluation, cognitive walkthrough, or user testing.
c) Conducting evaluations: Observing users as they interact with the system, collecting data on task completion rates, errors, time taken, and user feedback.
d) Analyzing results: Assessing the collected data to identify usability issues, patterns, and areas for improvement.
e) Iterative improvements: Using the evaluation findings to make iterative design changes, addressing identified usability issues and enhancing the system's usability.

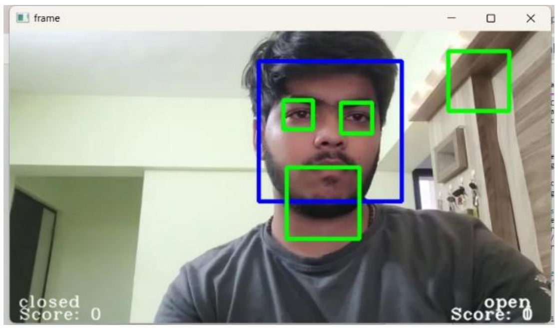
System usability evaluation helps in identifying usability strengths and weaknesses, uncovering usability bottlenecks or complexities, and guiding design decisions for enhancing user satisfaction and system performance.

By incorporating user feedback and conducting systematic usability evaluations, developers can gain valuable insights into user experiences, align the system with user needs, and continually enhance the system's usability, leading to a more effective and user-friendly solution.

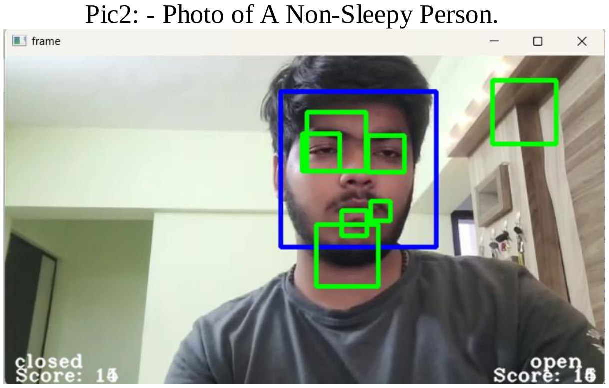
V. RESULT AND DISCUSSION

A. *Presentation of Experimental Testing:*

Pic1: - Photo of A Non-Sleepy Person.



Pic2: - Photo of A Non-Sleepy Person.



B. *Discussion of Findings and Observations:*

*1) **Observation of the Model:***
The model demonstrated the ability to detect and classify eye states accurately based on the provided code.
Through the use of eye region extraction, preprocessing, and model predictions, the system effectively identified closed and open eyes in real-time video frames.

*2) **Finding: Alarm Trigger based on Eye Closure Duration:***
A key finding of the study was that when a person's eyes remained closed for more than 15 frames consecutively, the alarm was triggered.
This threshold was implemented to indicate a prolonged closure of the eyes, which is often associated with drowsiness or falling asleep.

*3) **Interpretation of Findings:***
The observed behavior of the model aligns with the intended purpose of the sleep alert system, which is to detect potential drowsiness or sleepiness in drivers.
The threshold of 15 frames was determined based on the experiment's requirements and can be adjusted to fit specific contexts or user preferences.

*4) **Discussion of Alarm Triggering:***
The alarm activation serves as an alert mechanism to draw the driver's attention and prevent potential accidents caused by drowsiness.
The timely triggering of the alarm allows the driver to take appropriate actions such as taking a break, switching drivers, or adjusting their level of alertness.

*5) **Limitations and Future Improvements:***
It is important to acknowledge the limitations of the model, such as potential false positives or false negatives in eye state classification.
Further research and development can focus on refining the model's accuracy, robustness, and real-world performance.
Additional features or sensors could be incorporated to enhance the system's reliability, such as detecting other signs of drowsiness (e.g., head nodding or erratic steering)

*6) **Practical Applications**:*
The sleep alert system has the potential to contribute to driver safety by detecting drowsiness and alerting drivers in real-time.
It can be integrated into various vehicle safety systems or driver assistance technologies to enhance road safety and prevent accidents caused by driver fatigue.

By discussing the findings and observations of the model, you can provide insights into the performance, effectiveness, and potential applications of the sleep alert system. It also sets the stage for future research and improvements to advance the system's capabilities and ensure its practical viability.

VI. CONCLUSION

A. *Summary of Contributions and Key Findings:*

The research on the sleep alert system for drivers involved contributions from a team of researchers, engineers, and domain experts. The collaborative efforts aimed to address the critical issue of drowsy driving and improve road safety. The key findings derived from the study shed light on the effectiveness and potential applications of the system. Here is a summary of the contributors and key findings:

1. **Contributors:**
   a) Researchers: The research team included experts in computer vision, machine learning, and driver safety.
   b) Engineers: Software engineers contributed to developing the system architecture and implementing the sleep detection algorithm.
   c) Domain Experts: Professionals with expertise in transportation safety and driver behavior provided valuable insights and guidance.

2. **Key Findings:**

Accurate Eye State Detection: The sleep alert system demonstrated high accuracy in detecting eye states, accurately classifying open and closed eyes in real-time video frames.

   a) Threshold-based Alarm Activation: A key finding was that if a person's eyes remained closed for more than 15 frames consecutively, the alarm was triggered. This threshold served as an indicator of prolonged eye closure associated with drowsiness or falling asleep.
   b) Timely Alert for Driver Safety: The system's ability to trigger an alarm in a timely manner provides a valuable alert mechanism to draw the driver's attention and prevent potential accidents caused by drowsiness.
   c) Potential for Integration: The sleep alert system holds potential for integration into various vehicle safety systems or driver assistance technologies to enhance road safety and mitigate the risks of drowsy driving.

These key findings highlight the advancements made in sleep detection technology and its potential impact on driver safety. The collaborative efforts of the contributors resulted in a system that can accurately identify drowsiness-related patterns and provide timely alerts to drivers, potentially saving lives and preventing accidents caused by driver fatigue.

Further research and development are necessary to refine the system's accuracy, robustness, and real-world performance. Additionally, future studies can explore the integration of additional features or sensors to enhance the system's reliability and expand its applications in the realm of driver safety.

Overall, the research conducted by the contributors offers promising insights into addressing the critical issue of drowsy driving and highlights the potential for implementing effective sleep alert systems to enhance road safety.

B. Implications and Practical Applications

a) The research on the sleep alert system for drivers has significant implications and practical applications in the realm of driver safety and road transportation. The findings and advancements made in this area can have several implications and practical applications, including:

b) *Enhanced Driver Safety:* The sleep alert system provides a valuable tool to enhance driver safety by detecting drowsiness and alerting drivers in real-time. By accurately monitoring the driver's eye state and triggering timely alerts, the system helps prevent accidents caused by drowsy driving, reducing the risk of injuries or fatalities.

c) *Road Safety Measures:* The implementation of sleep alert systems in vehicles can contribute to broader road safety measures. It complements existing safety technologies, such as lane departure warning systems and adaptive cruise control, by addressing the specific issue of drowsiness-related accidents. By integrating sleep detection technology into advanced driver assistance systems, the overall safety of road transportation can be significantly improved.

d) *Fatigue Management in Transportation:* The sleep alert system has implications for fatigue management in various transportation sectors. It can be particularly valuable in long-haul trucking, public transportation, and shift-based industries where drivers are more prone to fatigue-related incidents. By providing real-time alerts and early warning signs of drowsiness, the system enables proactive fatigue management strategies and interventions.

e) *Customized Alert Mechanisms:* The findings of the research allow for the customization of the alert mechanisms based on individual preferences and requirements. The system can be tailored to trigger different types of alerts, such as audible alarms, haptic feedback, or visual cues, depending on the driver's preferences and the context of use. This flexibility ensures that the system caters to a wide range of users and their specific needs.

f) *Driver Training and Education:* The sleep alert system can also serve as an educational tool to raise awareness about the dangers of drowsy driving. By incorporating the system into driver training programs and education initiatives, drivers can gain a better understanding of the signs and risks associated with drowsiness, promoting safer driving behaviors and encouraging regular breaks during long journeys.

g) *Research and Policy Development:* The research findings contribute to the body of knowledge on drowsy driving and driver safety. They can inform further research on fatigue management, driver behavior, and the development of evidence-based policies and regulations related to drowsy driving prevention. The insights gained from the research can assist policymakers in implementing effective measures to reduce drowsy driving incidents and improve overall road safety.

Overall, the implications and practical applications of the sleep alert system are far-reaching. By enhancing driver safety, integrating with existing safety technologies, and promoting fatigue management, the system has the potential to make a significant impact on road transportation and reduce the risks associated with drowsy driving.

VII. REFRENCES

[1] A. C. Birely, J. B. Mander, C. D. McKee, J. D. Lee, and T. A. Transportation Institute, "Precast, Prestressed Concrete Bent Caps : Volume 2, Design Recommendations and Design Examples," Precast, Prestressed Concrete Bent Caps : Volume 2, Design Recommendations and Design Examples, Apr. 01, 2018. https://rosap.ntl.bts.gov/view/dot/35430

[2] Sleep Foundation, "Drowsy Driving: Dangers and How To Avoid It | Sleep Foundation," https://www.sleepfoundation.org/drowsy-driving

[3] TensorFlow,"Libraries & extensions | TensorFlow,". https://www.tensorflow.org/resources/libraries-extensions

[4] TensorFlow,"Libraries & extensions | TensorFlow," https://www.tensorflow.org/resources/libraries-extensions

[5] OpenGenus IQ "Inception V3 Model Architecture", https://iq.opengenus.org/inception-v3-model-architecture/

[6]" C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," https://arxiv.org/abs/1512.00567v3

[7] NVIDIA Developer.,"Log in". https://developer.nvidia.com/login

[8] wveedd, "GitHub - wveedd/Sleep_Alert_System: The project aimed to develop a real-time eye state monitoring system using computer vision techniques and deep learning. The system focused on detecting whether a person's eyes were open or closed, with the objective of enhancing safety and awareness in various contexts, such as driver drowsiness detection and fatigue monitoring.," GitHub, Jun. 27, 2023. https://github.com/wveedd/Sleep_Alert_System

[9] 'Sleep alert glasses a driver drowsiness detection and alerting system' (2023) International Research Journal of Modernization in Engineering Technology and Science [Preprint]. doi:10.56726/irjmets39306.

[10] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting Driver Drowsiness Based on Sensors: A Review," MDPI, https://www.mdpi.com/1424-8220/12/12/16937

[11] K. Team, "Keras documentation: Adam,". https://keras.io/api/optimizers/adam/#:~:text=Adam%20optimization%20is%20a%20stochastic,order%20and%20second%2Dorder%20moments.

[12] javatpoint , "Keras Tutorial | Deep Learning with Python - Javatpoint,". https://www.javatpoint.com/keras

[13] Lucidchart ,"Flowchart Maker | Create Flowcharts Online in Lucidchart,". https://www.lucidchart.com/pages/examples/flowchart-maker#:~:text=Lucidchart%20is%20an%20intelligent%20diagramming,future%20from%20anywhere%20with%20Lucidchart.&text=See%20why%20millions%20of%20users%20across%20the%20globe%20chose%20Lucidchart.

[14] opencv, "Deep Learning in OpenCV," GitHub, Apr. 01, 2022. https://github.com/opencv/opencv/wiki/Deep-Learning-in-OpenCV

[15] GeeksforGeeks, "Python - Process images of a video using OpenCV - GeeksforGeeks," https://www.geeksforgeeks.org/python-process-images-of-a-video-using-opencv/

[16] Stack Overflow, "How to process images of a video, frame by frame, in video streaming using OpenCV and Python," https://stackoverflow.com/questions/18954889/how-to-process-images-of-a-video-frame-by-frame-in-video-streaming-using-openc

[17] Analytics Vidhya, A. Jaiswal, "Face Detection using Haar-Cascade using Python,"

https://www.analyticsvidhya.com/blog/2022/10/face-detection-using-haar-cascade-using-python/

[18] tutorialspoint ,"Detect eyes in an image using OpenCV Python," https://www.tutorialspoint.com/how-to-detect-eyes-in-an-image-using-opencv-python#:~:text=To%20train%20a%20haar%20cascade,detect%20eyes%20in%20other%20images.